Introduction

This document updates the current status of our primary single client, the head of Data Science for our corporation. A 20-minute interview reviewed his fragmented awareness of his system architecture and identified features that may improve his team's satisfaction. The most salient point is that despite his heavy dependence, our client has cursory relationship with batch processing except to say that when batches interfere with his access to complete and cleaned data, he prepares the data himself. In terms of current architecture, our client indicates there's a need for greater integration to which we understand points to data sources, and thusly points back to batch processing.

Current system overview

Our corporation's data science team provides service internally and externally with regular and ad hoc analysis. Most if not all ad hoc analysis is for internal purposes and thus carries a high value for decision making. Most external analysis in the form of financial and corporate market intelligence which has reputation consequences and so timeliness and accuracy are important. The former may come at odd times and thus intersect with batch processing, while the latter is more systematic, around which a batch processing schedule can be fitted.

Their data structured and unstructured format are internally derived from back-end servers, data lakes, and a data warehouse. Data might be the product of OLTP but is consumed from a data warehouse for analysis (OLAP). For these data forms, access is via SQL (RDB), No SQL and ElasticSearch (document format). Scraped into data lakes are data such as climate and environmental data, social media and data from other public data sources. Facebook and

Google are a public data source but also partner data sources. Some of this data is treated and other is schema on read, simply collected without an intended purpose or format or a time-frame for format development. No mention was made of the sushi principle, but there are other varied internal data consumers and our client is a data veteran from IBM research and close associate of Yann LeCun. So, I would imagine that data lakes serve a useful purpose when a pipe is opened up.

Map reduce and HDFS are not as prevalent in systems as Spark not streaming and Spark streaming. Possibly Hadoop MapReduce is employed for their Lucene-indexed document databases accessed via ElasticSearch. HDFS provides the facility to bring data in without considering it's format or plan for format which suits our clients web scraping needs. HDFS may also be employed for their market data services as transaction processing systems bring raw price data into an MPP data warehouses and batch cleanups are run nightly for consumption the next day.

Hardware is clustered in a distributed network format. AWS is used for computation only, storage is all hosted internally for security purposes. Windows dominates and to a lesser extent Linux is used.

Our client has zero knowledge about the finer points of architecture, beyond the distributed network, nature. As an experienced data scientist, his tool kit includes shell via Windows for data access to internal frameworks, but more dedicated platforms such as ElasticSearch, SQL, Jupyter are widely used by his team.

Current batch processing occurs outside normal working hours, and scheduling depends on the data. Some are daily, some weekly or monthly. Market moving price data is batch processed daily or intra daily and may take as little as 30 minutes to complete. Some market price data where low latency is key may be nearline processed. Economic and corporate reports might be completed weekly or monthly and corporate reports such as earnings can be processed quarterly. When asked why batches are scheduled when they are scheduled, our client said that timing is up to the data team. Our client is a new joiner in the last 2 years. Other more knowledgeable internal consumers may comprehend and interact with scheduling better than he.

Limitations of current system / recommendations for next steps

There is clearly some value to be added in comprehending his need for data integration. In the same context he mentioned excessive duplication where the same data is available in separate systems. As a formal scientist, he may offer a clues to ways that our data volumes can be reconfigured to appeal to more users. That he steps around batch processing clearly indicates he interacts with its outcome, but that scheduling can be improved. That he is unaware how scheduling is done and his expressed interest for integration would suggest further meetings to explore which data can be integrated, where duplication exists. Can data lakes be mined for more structured forms via batch processing? Aware of Spark, as our dataflow engine, he must also know how optimization and reduced materialization enable us to assemble data in a more flexible more Unix like fashion with fault tolerance based on resilient distributed dataset abstraction. Are there larger data sets that we can process with an advantage versus our competitors due to this set up? Are there ways we can leverage Spark's

bulk synchronous parallel model of computation to process machine learning graphs. Can those linkages batch process his climate or social network data overnight instead of weekly and link with nearline processed market data the next day to produce valuable daily reusable implementations for external clients? Would these fault tolerances resulting reliability make our corporation the go-to house for linked data?

In short, we need more meetings with this fellow to explore the value of integrating existing data, employing unused data and reconfiguring our batch processes to address those data in a scheduled fashion that generates a competitive advantage for our corporation.